

当我们在SQL里进行排序的时候，如何才能使用索引？

之前我们已经给大家讲解了在SQL里使用where语句进行数据过滤和筛选的时候，在where语句里要如何写才能用上我们建立好的索引，其实无论是哪条规则，总之，尽可能就是从联合索引最左侧的字段开始去使用，就能用上索引树！

那么今天我们来讲一下，当我们的SQL语句里使用order by语句进行排序的时候，如何才能用上索引呢？

通常而言，就我们自己想象一下，假设你有一个select * from table where xxx=xxx order by xxx这样的SQL语句，似乎应该是基于where语句通过索引快速筛选出来一波数据，接着放到内存里，或者放在一个临时磁盘文件里，然后通过排序算法按照某个字段走一个排序，最后把排序好的数据返回。

但是这么搞通常速度有点慢，尤其是万一你要排序的数据量比较大的话，还不能用内存来排序，如果基于磁盘文件来排序，那在MySQL里有一个术语，叫做filesort，这速度就比较慢了。

通常而言，咱们尽量是最好别这么搞，尤其是类似于select * from table order by xx1,xx2,xx3 limit 100这样的SQL语句，按照多个字段进行排序然后返回排名前100条数据，类似的语句其实常见于分页SQL语句里，可能需要对表里的数据进行一定的排序，然后走一个limit拿出来指定部分的数据。

你要是纯粹把一坨数据放到一个临时磁盘文件里，然后直接硬上各种排序算法在磁盘文件里搞一通排序，接着按照你指定的要求走limit语句拿到指定分页的数据，这简直会让SQL的速度慢到家了！

所以通常而言，在这种情况下，假设我们建立了一个INDEX(xx1,xx2,xx3)这样的联合索引，这个时候默认情况下在索引树里本身就是依次按照xx1,xx2,xx3三个字段的价值去排序的，那么此时你再运行select * from table order by xx1,xx2,xx3 limit 100这样的SQL语句，你觉得还需要在什么临时磁盘文件里排序吗？

显然是不用了啊！因为他要求也不过就是按照xx1,xx2,xx3三个字段来进行排序罢了，在联合索引的索引树里都排序好了，直接就按照索引树里的顺序，把xx1,xx2,xx3三个字段按照从小到大的值获取前面100条就可以了。

然后拿到100条数据的主键再去聚簇索引里回表查询剩余所有的字段。

所以说，在你的SQL语句里，应该尽量最好是按照联合索引的字段顺序去进行order by排序，这样就可以直接利用联合索引树里的数据有序性，到索引树里直接按照字段值的顺序去获取你需要的数据了。

但是这里有一些限定规则，因为联合索引里的字段值在索引树里都是从小到大依次排列的，所以你在order by里要不然就是每个字段后面什么都不加，直接就是order by xx1,xx2,xx3，要不然就都加DESC降序排列，就是order by xx1 DESC,xx2 DESC,xx3 DESC。

如果都是升序排列，直接就从索引树里最小的开始读取一定条数就可以了，要是都是降序排列，就是从索引树里最大的数据开始读取一定的条数就可以了，但是你不能order by语句里有的字段升序有的字段降序，那是不能用索引的。

另外，要是你order by语句里有的字段不在联合索引里，或者是对order by语句里的字段用了复杂的函数，这些也不能使用索引去进行排序了。

所以说，今天的内容学完，那大家对于SQL语句的order by排序如何使用索引直接提取数据就心里有数了，其实这一讲内容是很实用的，因为我们平时写一些管理系统最常见的分页语句的时候，往往就是select * from table order by xxx limit xxx,xx这样的写法，按照某个字段自动排序，同时提取每一页的数据，所以如果你可以在排序用上索引，那么可以说你的性能就会很高。

End